INCORPORATING HIGH PRESSURE PHASE EQUILIBRIA IN SIMULATION OF DISTILLATION COLUMNS

A Thesis Submitted
in Partial Fulfilment of the Requirements
for the Degree of

MASTER OF TECHNOLOGY

by

Sanjay Kumar Singh

to the

DEPARTMENT OF CHEMICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY , KANPUR

January 1996

CONTENTS

		Page
ABSTRACT		
1.	INTRODUCTION	1
2	LITERATURE REVIEW	4
3.	MODELLING OF MULTISTAGE MULTICOMPONENT SEPERATION COLUMNS AND METHODS OF SOLUTION	7
	3.1 Modelling equations for single columns	
	3.2 Method of solution of Jacobian for single columns	
	3.3 Initial guesses, criterion of convergence and mapping functions	
	3.4. outline of computer program	
4.	THRMODYNAMIC ASPECTS OF EQUATION OF STATE APPROACH	28
	4.1 High pressure phase equilibria	
	4.2 Mixing Rules	
	4.3 Wong and Sandler's mixing rule with the Peng-Robinson equation od state	
5.	RESULTS AND DISCUSSION	34
	CONCLUSIONS	47
	NOMENCLATURE	48
	REFERENCES	51
	APPENDIX	53

1 2 MAY 1996

Acc No. A. 121497

CHE-1996-M-SON-INC.

A121497

9298 E6

ACKNOWLEDGEMENTS

: am grateful to Prof.D.P.Rao and Prof.Y.V.C.Rao for their help nd guidance in this work.I am also indebted to Mr.Adhip Sen Gupta or helping me in his own ways.

I am thankful to Mr. Pavitra shandilia for his ooperation and helpful discussions at all the stages of this ork.

Thanks are due to my classmates, Chavan, abhijit, urendra,Naveen who have contributed to my cherished stay at IIT anpur.

Sanjay Kumar Singh

CERTIFICATE

It is certified that the present work entitled , *"Incorporating high pressure phase equilibria in simulation of distillation columns"* , has been carried out by **Mr. SANJAY KUMAR SINGH** under my supervision and that it has not been submitted elsewhere for a degree.

January, 1996

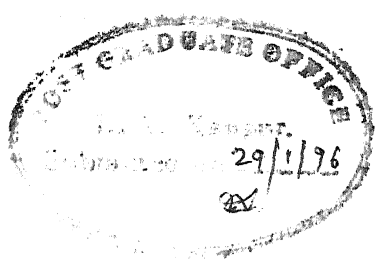
(Dr. Y.V.C.RAO)

Professor

Department of Chemical Engineering

Indian Institute of Technology

Kanpur .



ABSTRACT

The steady state simulation of multistage, multicomponent separators requires the solution of a set of nonlinear algebraic equations. The Newton-Raphson technique is commonly used to solve the set of equations. The Napthali-Sandholm method for single separation columns gives rise to a block-tridiagonal Jacobian. Most of the blocks are highly sparse. A modified Gaussian elimination has been developed to solve the jacobian Many separation problems occur at high pressures where gamma-phi approach will lead to errors.For those cases the Equation of state approach has been incorporated .The latest mixing rule given by Wong and Sandler has been used with Peng-Robinson equation of state to calculate K in this present work. Computer programs have been written implementing the modified VLE calculations and successfully tested with absorption and distillation problems.

CHAPTER : 1
INTRODUCTION

Separation processes like absorption, extraction, distillation etc. are widely used in chemical process industries to separate binary or multicomponent mixtures. Separation units are often the major factor in plant capital and operating costs. Therefore design, modelling and simulation of these units are extremely important. In this work, stagewise contact separation columns have been considered.

The modern day computers have drastically reduced the manual effort involved in solving separation problems in a rigorous manner. Many approaches have been proposed in the literature for efficient solution of the nonlinear algebraic equations involving mass balance, heat balance and equilibrium relations which arise in the modelling of the separation columns. The simultaneous convergence method proposed by Naphtali and Sandholm [5] is a well established procedure to simulate single separation columns.

In the Naphtali-Sandholm method, the set of nonlinear algebraic equations are solved by the Newton-Raphson technique. The iterative solution involves a Jacobian, which is a large and sparse matrix. In case of single separation columns the Jacobian has a block-tridiagonal structure.Most of the constituent blocks of Jacobian are highly sparse. The large and sparse structure of the Jacobian can be exploited in solving the linear equations efficiently. The Newton-Raphson method may fail to converge to solution if the initial guess is not within the domain of convergence and more over, only one solution can be obtained from a particular

starting point even if there are multiple solutions. These limitations of the Newton-Raphson technique can be overcome by the use of homotopy continuation methods. These methods convert the nonlinear algebraic equations to a set of coupled nonlinear ordinary differential equations with initial conditions. These equations are solved by predictor - corrector methods which trace out the solution path starting from the initial condition. The actual solution procedure involve the solution of large and sparse matrices for every predictor-corrector step. The structure of these matrices are that of the Jacobian of the Newton-Raphson method with one additional row and a column at the end. Efficient solution of the matrices are a crucial part of the solution procedure.

The initial guess for Newton-Raphson method can be generated in various ways. However, there is no single scheme which can ensure convergence for all types of separation problems. Recently, Venkata Ramana[7] proposed a simple initialisation scheme based feed conditions.

The objective of the present work is :

To incorporate high pressure phase equilibria based on equation of state approach.

In chapter 2 of the present work, a brief review of the literature avaliable on the relevant topics is presented. Chapter 3 contains the modelling equations and the methods of solution by Newton-Raphson procedure using the proposed schemes, for single columns . Chapter 4 describes the theoretical aspects of

Equation of State approach. Finally, results and discussions are presented in Chapter 5.

CHAPTER : 2

LITERATURE REVIEW

The steady state simulation of a stagewise contact separation column involves the determination of the temperatures and component flow rates for every stage and at the ends of the column under specified conditions. The basic procedure is to solve the component mass balances, equilibrium relations and enthalpy balances for all the stages. The large number of variables and complexity of the equations have led to many different solution techniques. A good review of the previous work has been presented by Venkata Ramana [7].

In a classic paper in 1971, Naphtali and Sandholm[5] proposed simultaneous convergence scheme for all the stage temperatures and component flow rates. They grouped by generating the equations in a stage by stage fashion. The Murphree stage efficiencies were included in the equlibrium relations.

The set of nonlinear equations are solved by using the Newton-Raphson method.The Jacobian which arises in the Newton -Raphson procedure has a large and sparse structure. In case of single columns, the Jacobian has a block tri-diagonal structure.

A brief review of the suggested methods of solving the Jacobian is presented below.

1) Single column : Naphtali and Sandholm used the standard Gauss elimination scheme to solve the tridiagonal Jacobian. Henley and Seader [3] extended the Thomas algorithm for

tridiagonal matrices to the case of the block-tridiagonal Jacobian. Their procedure, known as the block Thomas algorithm, is a popular method to solve the Jacobian for single columns.

2. Continuation methods : The Newton-Raphson technique as applied to separation problems fail to converge to a solution if the initial guess is not in the domain of convergence. Also the method produces only one solution from a given starting point. In cases where the Newton-Raphson method fails to converge to a solution or where multiple solutions are being seeked, the continuation methods are employed. Jelinek et al. [9] appear to be the first to employ the concept of continuation to separation problems. They obtained solutions for multicomponent multistage separation with constant flow rate using reflux ratio as the continuation parameter. As mentioned by Wayburn and Seader [8], attempts have been made to apply continuation to separation problems using different homotopy functions, attempting stabilisation of the ODE-IVP formed from the algebraic equations and also by path parametrisation. They suggested an affine homotopy which prevents the solution path doubling back to the initial guess value in place of the simple Newton's homotopy. Wayburn and Seader [10] obtained multiple solutions for interlinked distillation column problems and showed applicability of continuation in cases where the Newton-Raphson method failed. A recent review of the previous work can be found elsewhere [7]. Recently Wong and Sandler[11] proposed a newmixing rule.This new mixing rule with Peng-Robinson Equation of State has been used to calculate thr equilibrium constants[K] which is required when

one uses the Murphree efficiency as the equilibrium relation.

CHAPTER : 3

MODELLING OF MULTISTAGE MULTICOMPONENT SEPARATION COLUMNS AND METHODS OF SOLUTION

In this chapter the modelling equations for multistage multicomponent separators and their methods of solution are presented. The model proposed by Naphtali and Sandholm [5] for single columns has been followed with certain modifications as discussed by Venkata Ramana [7]. A method of solution of the Jacobian using a modified Gaussian elimination scheme has been presented.

A generalised modified Gaussian elimination scheme has also been proposed to handle the different possible configurations of the Jacobian.

3.1 Modelling Equations for Single Columns :

A counter-current steady state vapour-liquid separator consisting of N stages and separating a C component mixture is considered. The stages of the separator are numbered from the top to bottom. A schematic representation of a general stage j of the separator is depicted in Figure-1. As can be seen from Figure -1 the liquid and vapour leaving the stage are denoted by L_j and V_j respectively, the vapour from the lower stage and the liquid from the upper stage are denoted by V_{j+1} and L_{j-1} repectively, the stage has a feed f_j and the liquid and vapour side-streams leaving the stage are denoted by $s_j L_j$ and $S_j V_j$ respectively. A heat input/withdrawal of q_j is also considered.

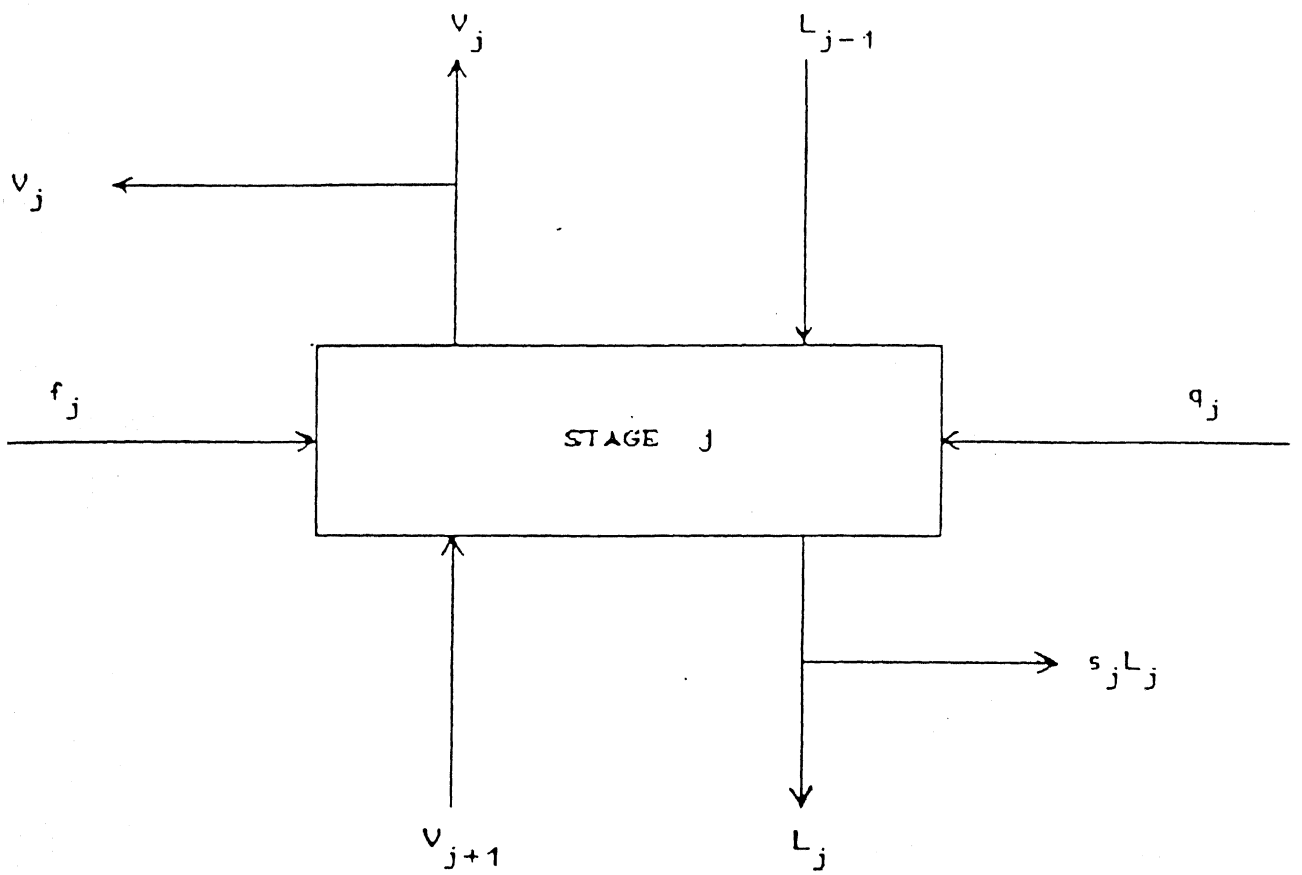


FIG.1 A GENERAL STAGE IN A SINGLE COLUMN

The modelling equations for this general stage are the component material balances, equilibrium relations and the enthalpy balance. These equations when expressed as discrepancy functions appear as follows :

(a) Component Mass Balance :

$$M_{j,i} = l_{j-1,i} + v_{j+1,i} - (1+s_j)l_{j,i} - (1+S_j)v_{j,i} + f_{j,i} \qquad \text{.....(3.1.1)}$$

where , $1 \le i \le C$, & , $1 \le j \le N$

(b) Equilibrium Relations :

$$Q_{j,i} = \frac{\eta_{j,i}K_{j,i}}{L_j}l_{j,i} - \frac{v_{j,i}}{V_j} + (1-\eta_{j,i})\frac{v_{j+1,i}}{V_{j+1}} \qquad \text{.....(3.1.2)}$$

$$\eta_{j,i} = \frac{y_{j,i} - y_{j+1,i}}{K_{j,i}\,x_{j,i} - y_{j+1,i}} \qquad \text{.....(3.1.3)}$$

where , $1 \le i \le C$, & , $1 \le j \le N$ and $\eta_{j,i}$ is the Murphree efficiency for component i at stage j.

(c) Enthalpy Balance :

$$E_j = \sum_1^C h_{j-1,i}l_{j-1,i} + \sum_1^C H_{j+1,i}v_{j+1,i} - (1+s_j)\sum_1^C h_{j,i}l_{j,i}$$

$$-(1+S_j)\sum_1^C H_{j,i}v_{j,i} + \sum_1^C h_{F_{j,i}}f_{j,i} + q_j$$

$1 \le j \le N$ and $1 \le i \le C$

.....(3.1.4)

S_j and s_j are the fractions of the vapour and liquid side-streams from the j-th stage respectively.

The above equations form a set of (2C+1) equations in terms of (2C+1) variables for each stage and hence a total of N(2C+1) equations in N(2C+1) unknowns for the entire column. The above equations can be written in a compact form as

$$\mathbf{F(X) = 0} \qquad \ldots.(3.1.5)$$

where

$$\mathbf{F} = (F_1, F_2, \ldots\ldots, F_j, \ldots\ldots F_N)^T \qquad \ldots.(3.1.6)$$

$$F_j = (M_{j,1}, \ldots, M_{j,C}, Q_{j,1}, \ldots, Q_{j,C}, E_j)^T \qquad \ldots.(3.1.7)$$

$$\mathbf{X} = (X_1, X_2, \ldots\ldots, X_j, \ldots., X_N)^T \qquad \ldots.(3.1.8)$$

$$X_j = (l_{i,1}, \ldots, l_{j,C}, v_{j,1}, \ldots, v_{j,C}, T_j)^T \qquad \ldots.(3.1.9)$$

As mentioned by Venkata Ramana [7], the equations and variables are ordered in such a way so as to get a matrix structure whose sparsity can be suitably exploited.

The Newton-Raphson method can be applied to obtain the solution of equation (3.1.5). The correction vector ΔX is given by

$$\Delta X^I = - \left[\left(\frac{\delta F}{\delta X} \right)^{-1} \right]^I F^I \qquad \ldots(3.1.10)$$

or,
$$\Delta X^I = - (J^{-1})^I F^I \qquad \ldots(3.1.11)$$

where J is the Jacobian matrix and I is the iteration number.

For the first iteration, an initial guess value of X, X^o has to be estimated and ΔX^1 calculated from equation (3.1.11). Then the I-th approximation of the variables is computed as

$$X^I = X^{I-1} + \theta \, \Delta X^I \qquad \ldots(3.1.12)$$

The value of θ can be chosen in such a way that the convergence is either accelerated or dampened. In the usual Newton-Raphson technique θ is taken as one. Henley and Seader [3] suggested a Fibonacci search to find an optimum θ for each

iteration. Venkata Ramana [7] adopted the Broyden's method to calculate θ at each iteration step and the same has been used in the present work.

3.2 Method of Solution of Jacobian for Single Column :

The structure of the Jacobian matrix obtained from the Naphtali-Sandholm formulation is block-tridiagonal in case of single columns.The structure of the Jacobian is given by

$$J = \begin{bmatrix} B_1 & C_1 & & & & & \\ A_2 & B_2 & C_2 & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & A_j & B_j & C_j & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & A_N & B_N & \end{bmatrix}$$

The structures of the A, B and C blocks are given by:

$$A_j = \begin{bmatrix} I_c & O_{cc} & O_c \\ O_{cc} & O_{cc} & O_c \\ X_c^T & O_c^T & X \end{bmatrix}$$

$$B_j = \begin{bmatrix} -(1+s_j)I_c & -(1+S_j)I_c & O_c \\ X_{cc} & X_{cc} & X_c \\ X_c^T & X_c^T & X \end{bmatrix}$$

$$C_j = \begin{bmatrix} O_{cc} & I_c & O_c \\ O_{cc} & X_{cc} & O_c \\ O_c^T & X_c^T & X \end{bmatrix}$$

where I_c is a C x C identity matrix, O_{cc} is a C x C null matrix, O_c a C dimensional null vector, X_{cc} is a C x C nonzero matrix, X_c is a C dimensional nonzero vector and X is a nonzero number. As can be seen, **A** is a highly sparse matrix, **C** is also sparse while **B** has few zeros.

The matrix equation involving the tridiagonal Jacobian can be solved by the block Thomas algorithm. However, a Gaussian elimination which takes into consideration the highly sparse structure of the Jacobian reduces the operation count significantly as will be shown later.

A method of solution for the linearised matrix equation $\mathbf{J}\ \Delta\mathbf{X} = -\mathbf{F}$ has been developed in the present work based on the Gaussian elimination technique. The proposed algorithm takes into account the sparse structure of the Jacobian and its constituent submatrices (blocks **A**, **B** and **C**) to reduce the operation count as well as storage requirements as compared to a full matrix or a block-tridiagonal matrix with filled up blocks.

The Gaussian elimination :

The Gaussian elimination technique is discussed briefly now. Consider a general matrix equation $\mathbf{M}\ \mathbf{x} = \mathbf{b}$, where **M** is a matrix of order m x m, **x** and **b** are m dimensional vectors. Let the

elements of **M**, **x** and **b** be as follows:

$$\mathbf{M} = \begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & \cdots & a_{mm} \end{vmatrix}$$

$$\mathbf{x} = |\bar{x}_1 \quad x_2 \quad \cdots \quad x_m \bar{}|^T \quad ; \quad b = |\bar{b}_1 \quad b_2 \quad \cdots \quad b_m \bar{}|^T$$

The augmented matrix of the above matrix equation is defined by the matrix $|\bar{\mathbf{M}} \; |b\bar{}|$. The Gaussian elimination procedure for this general matrix equation consist of two parts :

i) The forward elimination - A series of operations are carried out on the equations such that the **M** matrix is transformed into an upper-triangular matrix with diagonal elements as one. This is achieved by dividing the i-th row of the augmented matrix by the element at location (i,i), i.e., the pivot element (normalisation step). Then the resultant row is multiplied by the element at (l,i) of **M** and subtracted from the l-th row where $i+1 \leq l \leq m$ (reduction step). The above steps are carried out successively from i = 1 to i = m, ultimately yielding the upper-triangular matrix.

ii) The back substitution - The last element of the augmented matrix after the forward elimination is x_m. Now, the x_is are solved for with the help of known x_ls ($i+1 \leq l \leq m$) by moving up the rows of the augmented matrix.

The algorithm consists of the following steps :

Forward elimination

1. $a_{il} \longleftarrow a_{il}/a_{ii}$
2. $b_i \longleftarrow b_i/a_{ii}$ } $1 \le i \le m$; $i+1 \le l \le m$
3. $a_{kl} \longleftarrow a_{kl} - a_{ki}\, a_{il}$ } $i+1 \le k \le m$
4. $b_k \longleftarrow b_k - a_{ki}\, b_i$

Back substitution

5. $x_m = b_m$
6. $x_i = b_i - \sum_{l=i+1}^{m} a_{il}\, x_l$ $\quad m-1 \ge i \ge 1$

Modified Gaussian elimination for single column :

The Gaussian elimination has been suitably modified in the present work to exploit the block-tridiagonal structure of the Jacobian in case of a single column. Furthermore, the presence of zeros and ones in the individual blocks has been exploited to reduce the operation count, i.e., number of multiplications and divisions performed and also storage requirements.

Since all the diagonal elements (pivots) of the Jacobian are also the diagonal elements of **B** blocks, the forward elimination can be thought of as transforming **B** blocks to an upper-triangular form and reducing all the elements of the blocks directly below the **B** blocks to zeros. This way a block by block approach has been formulated. For the j-th **B** block, the only nonzero block directly below it is the block $\mathbf{A}_{j+1}$ because of the

tridiagonal structure of the Jacobian, and hence we have to operate only on the j-th and (j+1)-th rows of blocks for the j-th row. Also, there are no nonzero elements directly above C_{j+1}, and hence this block remains unaltered throughout the forward elimination of the j-th row and need not be considered. So, the blocks which get affected at the j-th step of the elimination are B_j , C_j , F_j , A_{j+1} , B_{j+1} and F_{j+1} . The A_j block reduces to zero at the (j-1)-th step and therefore is not present for forward elimination of the j-th row.

From the above discussion it follows that the operations for the j-th row of Jacobian will involve 4C+2 rows (2C+1 rows each for the j and (j+1)-th blocks) and 4C+3 columns (2C+1 columns corresponding to each of B_j and C_j and one corresponding to the F blocks). The operations can be easily understood by following the structure of the Jacobian at different stages during the forward elimination as shown in the following discussion.

Forward substitution :

For the first B block, the pivots are the elements (1,1) to (2C+1,2C+1). The structure of the blocks which are operated upon in the augmented Jacobian is

$$\left|\begin{array}{c|c|c} B_1 & C_1 & F_1 \\ \hline A_2 & B_2 & F_2 \end{array}\right| \equiv$$

	1	2	C	C+1	2C	2C+1	2C+2	3C+1	3C+2	4C+1	4C+2	4C+3
1	-1	0	..0	-1	.. 0	0	0	0	1 ...	0	0	x
2	0	-1	..0	0	.. 0	0	0	0	0 ...	0	0	x
	..	..	..	..	..	...	..	..	..	..	...	...
C	0	0	-1	0	..-1	0	0	0	0 ...	1	0	x
C+1	x	x	..x	x	.. x	x	0	0	x ...	x	0	x
	..	..	..	..	..	...	..	..	..	..	...	...
2C	x	x	..x	x	.. x	x	0	0	x ...	x	0	x
2C+1	x	x	..x	x	.. x	x	0	0	x ...	x	x	x
2C+2	1	0	..0	0	.. 0	0	-1	0	-1 ...	0	0	x
	..	..	..	..	..	...	..	..	..	..	...	...
3C+1	0	0	..1	0	.. 0	0	0-1		0 ...-1		0	x
3C+2	0	0	..0	0	.. 0	0	x x		x ... x		x	x
	..	..	..	..	..	...	..	..	..	..	...	...
4C+1	0	0	..0	0	.. 0	0	x x		x ... x		x	x
4C+2	x	x	..x	0	.. 0	x	x x		x ... x		x	x

where x represents nonzero numbers.

For the first C pivots the pivot row is divided by the quantity $(1+s_1)$ if $s_1 \neq 0$, if $s_1 = 0$ (as shown in the above diagram for the sake of simplicity) the divisions are not performed. Division is performed only on elements in columns C+i, 3C+1+i and 4C+3. As the rest of the elements of the row are zeros, they are not operated upon. The elements directly below the pivot element are reduced to zeros. Only the rows C+1 to 2C+1, 2C+1+i and 4C+2 are to be operated , since all the other rows below the pivot are already in the reduced form as can be verified from the structure of the individual blocks of the Jacobian. For the (2C+1+i)-th row no multiplication is involved as the element in the pivot column is unity. Again, only the elements in columns C+i, 3C+1+i and 4C+3 are affected for the rows operated upon.

For pivots (C+1,C+1) to (2C,2C), the pivot row is divided by the pivot element, the elements which are divided are in columns C+i+1 to 2C+1, 3C+2 to 4C+1 and 4C+3. Other elements of the row are zeros and hence not divided. The rows which are

considered for the reduction step are C+i+1 to 2C+1+i and 4C+2, the rest being already in the reduced form because of the A_{j+1} block structure. Also, only the elements in the columns C+i+1 to 2C+1, 3C+2 to 4C+1 and 4C+3 are affected.

For pivot (2C+1,2C+1), the (2C+1)-th row is divided by the (2C+1,2C+1)th element. The columns 2C+2 to 3C+1 are not considered, since the elements are zeros. The rows 2C+2 to 3C+1 and 4C+2 are reduced. Again, rows 3C+2 to 4C+1 are not operated upon due to the reduced structure they have in the original blocks. Elements corresponding to columns 2C+2 to 3C+1 are left out of the operations.

The above operations complete the first stage and the blocks under consideration now have the following structure

$$\left|\begin{array}{c|c|c} B_1 & C_1 & F_1 \\ \hline A_2 & B_2 & F_2 \end{array}\right| \equiv$$

	1	2	C	C+1	2C	2C+1	2C+2	3C+1	3C+2	4C+1	4C+2	4C+3
1	-1	0	..0	x'..	0	0	0	0	x'...	0	0	x'
2	0	-1	..0	0 ..	0	0	0	0	0 ...	0	0	x'
	..	..	..	..	..	..		..		..	...	...
C	0	0	-1	0 ..	x'	0	0	0	0 ...	x'	0	x'
C+1	0	0	..0	1 ..	x'	x'	0	0	x'...	x'	0	x'
	..	..	..	..	..	..		..		..	...	...
2C	0	0	..0	0 ..	1	x'	0	0	x'...	x'	0	x'
2C+1	0	0	..0	0 ..	0	1	0	0	x'...	x'	x'	x'
2C+2	0	0	..0	0 ..	0	0	-1	0	x'...	x'	x'	x'
	..	..	..	..	..	..		..		..	...	...
3C+1	0	0	..0	0 ..	0	0	0-1		x'...	x'	x'	x'
3C+2	0	0	..0	0 ..	0	0	x x		x ...	x	x	x
	..	..	..	..	..	..		..		..	...	...
4C+1	0	0	..0	0 ..	0	0	x x		x ...	x	x	x
4C+2	0	0	..0	0 ..	0	0	x x		x'...	x'	x'	x'

The elements which have filled in the zeros of the blocks or had their values changed are denoted by x' in the above matrix.

It is seen that the B_1 block becomes upper-triangular with the diagonal elements as unity, the C_1 block retains its

structure except the identity C x C block which now becomes a diagonal matrix. The identity block remains unchanged if s_1 is equal to zero. The A_2 block becomes a null matrix after the operations. The B_2 block becomes more filled up, with its second C x C diagonal matrix becoming a full matrix and the zero vector also becoming filled up.

A point to be noted here is that the first C columns of the B_2 block remain unchanged because the first C columns of the block C_1 are all zeros. Also the (C+1)-th to 2C-th rows of B_2 are not affected because the corresponding A_2 block elements are zeros in the original matrix and remain so through out. These two features are retained through out the forward elimination.

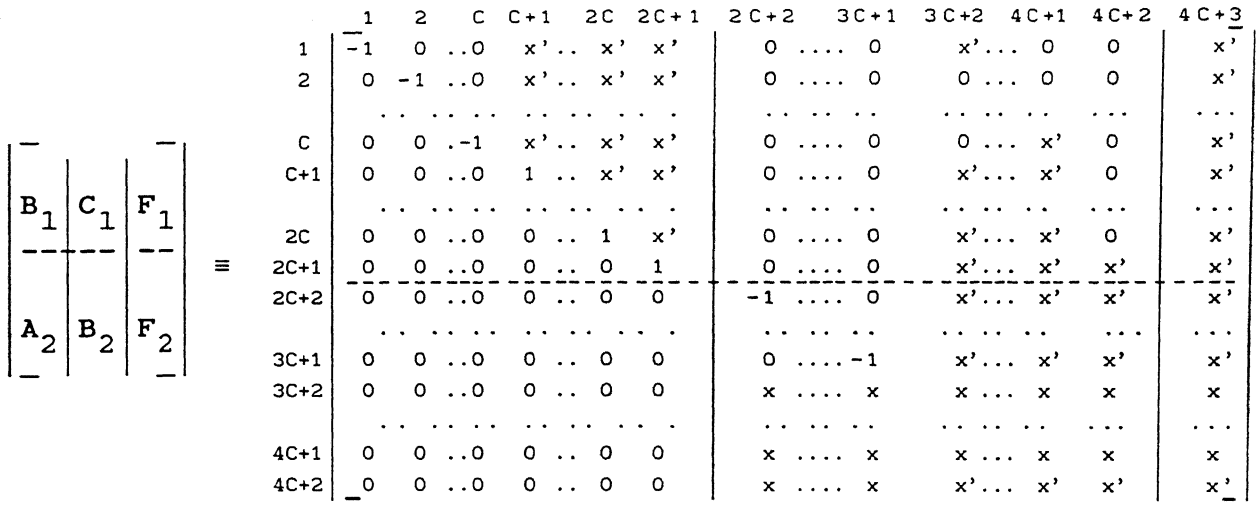
The nonzero elements of the blocks B_1 and C_1 and the vector F_1 are now stored for the back substitution.

The blocks B_2 , C_2 , F_2 , A_3 , B_3 and F_3 are to be considered for the second stage of the forward elimination.

The steps for pivots (1,1) to (C,C) are similar to those for the first row except that the elements in the columns C+1 to 2C+1 have to be operated upon in addition to the columns which are operated upon for the first row of blocks.

For pivots (C+1,C+1) to (2C+1,2C+1), the operations are the same as those for row 1.

The structure of the Jacobian after the second **B** block has been reduced is as shown below



Only the nonzero elements of the B_2 and C_2 blocks and the F_2 vector are stored for back substitution.

The point to note here is that the structure of the B_3 block now is same as the structure that the block B_2 had at the end of the forward elimination of the first row of blocks. The implication is that from here onwards the steps of the forward elimination will exactly be the same upto block B_{N-1} and the upper-triangular **B** blocks, its adjacent **C** blocks and the corresponding **F** blocks will be stored suitably at the end of the operations for each row of blocks.

For the last **B** block, i.e. B_N , the blocks involved in the operations are the B_N block itself and the F_N vector. No **C** or **A** blocks are to be considered in this case as B_N does not have any adjacent **C** block or lower **A** block to it. The operations involved are similar to those for the previous blocks, but there are no operations for row or column numbers greater than 2C+1 except the (4C+3)-th column corresponding to F_N . As before, the upper-triangular **B** block and the **F** block are stored in this case

for the back substitution.

Back substitution :

The upper-triangular matrix is now easily solved by back substitution starting with the last element of the N-th block. For any $\Delta X(j,i)$ solved for in the N-th block the values of the solved $\Delta X(j,k)$s are used, k going from i+1 to 2C+1. For $\Delta X(j,i)$, the i-th row of the N-th block is considered and $\Delta X(j,k)$s corresponding to the zeros in that row of the upper-triangular matrix are not considered for operations.

For blocks N-1 to 1, the ΔX values of only the (j+1)-th block have to be considered in addition to the j-th block for the j-th row of blocks. This is because of the block-tridiagonal structure of the Jacobian. For the i-th equation of the j-th block, all the (j+1)-th ΔXs corresponding to the nonzero elements of the (j+1)-th block in that row are considered. In addition, the $\Delta X(j,k)$s of the j-th block corresponding to the nonzero elements in the i-th row are also to be considered(for $i+1 \leq j \leq 2C+1$).

Implementation of the algorithm :

As mentioned earlier, the tridiagonal structure of the Jacobian clearly indicates that while performing the Gaussian elimination, only four blocks - the **B** block being reduced, the adjacent **C** block and the **A** and **B** blocks of the next row - are to be operated upon at a time in addition to the right hand side containing the discrepancy functions. The original **A**, **B** and **C** matrices need not be stored, only the four blocks which are being

operated upon can be generated at a time and the elements of the upper-triangular matrix formed after the Gaussian elimination can be stored along with the right hand side.

In the proposed algorithm, the **A**, **B** and **C** blocks and right hand side elements which are operated upon at a particular step are put in locations g(i,k) where i = 1,4c+2 and k = 1,4C+3. Then the Gaussian elimination from that step is performed. The final matrix elements from the j-th row (obtained from B_j & C_j blocks) are stored at locations $h^j(i,k)$ while the right hand side elements are stored at locations $d^j(k)$. Then the elements of the next row of blocks are shifted from their present locations to locations where the previous blocks of elements were kept,elements of the next generated blocks are put in the last (2C+1 x 4C+3) locations of g(i,k). This is continued upto the Nth stage to finally obtain the upper-triangular matrix which is then solved by the back substitution.

The elements of blocks B_j and C_j at the jth stage are g(m,k) where m goes from 1 to 2C+1 and k goes from 1 to 4C+2. Similarly the elements of A_{j+1} and B_{j+1} are kept at locations g(m,k), m going from 2C+2 to 4C+2 and k having the same ranges as before. The F_j and F_{j+1}vector elements are at locations g(m,4C+3) where m = 1,4C+2.

The steps of the Gaussian elimination :

The forward elimination :

To perform the forward elimination, a few basic operations are defined which can be used with different indices in the

various steps of the elimination. These operations are as follows

Operation 1 :

$$g(m,k) \longleftarrow g(m,k)/g(m,m)$$

Operation 2 :

$$g(m,k) \longleftarrow g(m,k) - g(l,n)$$

Operation 3 :

$$g(m,k) \longleftarrow g(m,k) + g(l,n)$$

Operation 4 :

$$g(m,k) \longleftarrow g(m,k) + g(m,l)\ g(l,k)$$

Operation 5 :

$$g(m,k) \longleftarrow g(m,k) - g(m,l)\ g(l,k)$$

Operation 6 :

$$h^j(m,k) \longleftarrow g(m,n) \qquad \text{(For } m= 1,2C+1;\ k= 1,C+1;\ n= C+1,2C+1 \ \&\ m= 1,2C+1;\ k= C+2,2C+2;\ n= 3C+2,4C+2\text{)}$$

Operation 7 :

$$d^j(m) \longleftarrow g(m,k) \qquad \text{(For } m = 1,2C+1\ ;\ k = 4C+3\text{)}$$

Operation 8 :

$$g(m,k) \longleftarrow g(2C+1+m,k)$$

(For $m = 1,2C+1$; $k = 1,2C+1$ & $4C+3$)

The steps of the Gaussian elimination now become :

Block B_1 ($j = 1$) :

Pivots (1,1) to (C,C) ($i = 1$ to C)

Operation 2 -- $m = C+1$ to $2C+1$ & $4C+2$; $k = C+i$; $l = m$; $n = i$

Operation 3 -- $m = C+1$ to $2C+1$; $k = 3C+1+i$; $l = m$; $n = i$

I.I.T. KANPUR

m = 2C+1+i ; k = 4C+3 ; l = m ; n = 4C+3

m = 4C+2 ; k = 3C+1+i ; l = m ; n = i

Operation 4 -- m = C+1 to 2C+1 & 4C+2; k = 4C+3 ; l = i

Pivots (C+1,C+1) to (2C,2C) (i = 1 to C)

Operation 1 -- m = C+i ; k = C+1+i to 2C+1 & 3C+2 to 4C+1 & 4C+3

Operation 5 -- m = C+1+i to 2C+2 & 4C+2 ; k = as above ; l = C+i

Operation 3 -- m = 2C+1+i ; k = as above ; l = C+i ; n = k

Pivot (2C+1,2C+1)

Operation 1 -- m = 2C+1 ; k = 3C+2 to 4C+3

Operation 5 -- m = 2C+2 to 3C+1 & 4C+2 ; k = as above ; l = 2C+1

Operations 6, 7 and 8 are also done.

Blocks B_2 to B_{N-1} (j = 2 to N-1) :

Pivots (1,1) to (C,C) (i = 1 to C)

Operation 4 -- m = C+1 to 2C+1 ; k = C+1 to 2C+1 & 4C+3 ; l = i

m = 2C+1 ; k = as above ; l = i

Operation 3 -- m = 2C+1+i ; k = as above ; l = m ; n = i

m = C+1 to 2C+1 ; k = 3C+1+i ; l = m ; n = i

m = 4C+2 ; k = 3C+1+i ; l = m ; n = i

Pivots (C+1,C+1) to (2C,2C) (i = 1 to C)

Operation 1 -- m = C+i ; k = C+1+i to 2C+1 & 3C+2 to 4C+1 & 4C+3

Operation 5 -- m = C+1+i to 3C+1 & 4C+2 ; k = as above ; l = C+i

Operation 3 -- m = 2C+1+i ; k = as above ; l = C+i ; n = k

Pivot (2C+1,2C+1)

Operation 1 -- m = 2C+1 ; k = 3C+2 to 4C+3

Operation 5 -- m = 2C+2 to 3C+1 & 4C+2 ; k = as above ; l = 2C+1

Operations 6, 7 and 8 are also to be done.

Block B_N (j = N) :

Operations are similar to those with blocks B_2 to B_N, but the steps involving m & k > 2C+1, except k = 4C+3, are not to be performed.

The back substitution :

The basic operations for the back substitution are :

Operation 1 :

$$d^j(k) \leftarrow\!\!- d^j(k) - \sum_m h^j(m)\ \Delta X(l,i)$$

Operation 2 :

$$d^j(k) \leftarrow\!\!- d^j(k) - \Delta X(j+1,C+k)$$

Operation 3 :

$$d^j(k) \leftarrow\!\!- d^j(k) + \Delta X(j+1,C+k)$$

Operation 4 :

$$d^j(k) \leftarrow\!\!- - d^j(k)$$

Operation 5 :

$$\Delta X(j,k) = d^j(k)$$

Now the back substitution steps can be written in terms of the above operations as follows :

Last (Nth) B block (j = N)

$\Delta X(j,2C+1)$: k = 2C+1

Operation 6

$\Delta X(j,2C)$ to $\Delta X(j,C+1)$: k = 2C to C+1

Operation 1 -- m = C+1 to k+1-C, l = N, i = C+m

Operation 6

$\Delta X(j,C)$ to $\Delta X(j,1)$: k = C to 1

Operation 1 -- m = 1 to C+1, l = N, i = C+m

Operation 4

Operation 6

(N-1)th to second B blocks (j = N-1,2)

$\Delta X(j,2C+1)$: k = 2C+1

Operation 1 -- m = C+2 to 2C+2, l = j+1, i = m-1

Operation 6

$\Delta X(j,2C)$ to $\Delta X(j,C+1)$: k = 2C to C+1

Operation 1 -- m = C+1 to k-C+1, l = j, i = C+m

Operation 1 -- m = C+1 to 2C, l = j+1, i = m

Operation 6

$\Delta X(j,C)$ to $\Delta X(j,1)$: k = C to 1

Operation 1 -- m = 1 to C+1, l = j, i = C+m

Operation 2

Operation 4

Operation 6

First B block (j = 1)

$\Delta X(j,2C+1)$: k = 2C+1

Operation 1 -- m = C+1 to 2C+1, l = j+1, i = m

Operation 6

$\Delta X(j,2C)$ to $\Delta X(j,C+1)$: k = 2C to C+1

Operations are same as in the case of k = 2C to C+1

for 2nd to (N-1)th rows

$\Delta X(j,C)$ to $\Delta X(j,1)$: k = C to 1

Operation 2

Operation 3

Operation 4

Operation 6.

3.3 Initial Guesses ,Criterion of convergence and Mapping Functions :

To ensure convergence with the Newton-Raphson method ,a good initial guess of the variables not too far away from the solution is required.Different initialisation schemes have been proposed in literature.Two of them are mentioned here.

1) Wayburn-Seader algorithm :In this method,proposed by Wayburn and Seader [10],stage temperatures are linearly interpolated between two or more guessed values.Usually the first and the last temperatures are guessed.The initial guesses of total flow rates are computed by considering constant molar overflow.The components flow rates are then determined using bubble point calculationa by the tridiagonal matrix method.

2) Initialisation with feed conditions :This is an extremely simple method of generating initial guess values.All the stage temperatures are set equal to the feed temperature. The total liquid and vapor flow rates are calculated by considering constant molar flow.The stream compositions are set equal to the feed composition.

The criterion of convergence can be selected from any of the various available methods.In the present work ,the convergence criterion,ϑ has been chosen following the recommendation of Henly and Seader [3] as :

$$\vartheta = N(2c+1) \left(\sum_{1}^{N} F_j^2\right) 10^{-10} \qquad (3.3.1)$$

Where F_j =total feed to stage j.According to them,this convergence criterion will provide results accurate to four or more significant digits .The Newton-Raphson iterations are stopped when the sum of the squares of the errors are less than or equal to ϑ,t is ,

$$\sum_{j=1}^{N} \left[(H_j)^2 + \sum_{i=1}^{C} \left((M_{i,j})^2 + (Q_{i,j})^2 \right) \right] \leq \vartheta$$

To avoid the possibilty of flow rates or temperatures becoming negative during iteration,a mapping function proposed by Henly-Seader [3] has been used.When negative flow rates and temperatures are encounterd,the variables are computed from the following relation

$$X^{I+1} = X^I \exp\left(\frac{\vartheta(\Delta X)^{I+1}}{X^I}\right)$$

instead of equation (3.1.12).

3.4 Outline of Computer Program :

Computer progarms have been written in FORTRAN 77 to implement the modified Gaussian elimination for single columns using the Newton-Raphson technique.The programs are divided into subroutines to perform the various steps.

The thermodynamic properties and feed specifications are read from separate input files.Parameters like reflux ratio,bottom flowrate or fraction of side streams are read from console.

There are provisions to calculate thermodynamic properties in sevaral ways.Like K can be calculated also by :1)As a temperature polynomial[12] 2)From the Wagner equation[6] 3) By gama-phi approa apart from the genaral Equation of State approach.The progarm allows the calculation of enthalpies in two ways :1)As a temperature polynomial 2)Using C_p as a temperature polynomial.

The partial derivatives ,which are the elements of the Jacobian , are calculated from analytical expressions in the program.The value of the fractional correction in the Newton-Raphson method can be calculated from either Broyden's formula or can be given by user at each iteration.

The problems solved using these progarms are reported in chapter 5.

CHAPTER : 4

HIGH PRESSURE PHASE EQUILIBRIA :EQUATION OF STATE APPROACH TO CALCULATE K.

Distillation is probably the most important separation process used in the chemical industry,where the coexisting phases are vapor and liquid.Hence reliable vapor liquid equilibrium data are required for the design of distillation columns.As it is not possible to get experimental VLE data for all multicomponent systems, so it is necessary to estimate the VLE data.

At low to moderate pressures ,the Gama-Phi approach can be used to get K.In this approach the fugacity coeficient is used to describe the vapor phase and the definition of activity coefficient is used to describe the liquid phase.At low pressures one can approximate the fugacity coefficient as equal to one in vapor phase.

Vapor liquid equilibria at high pressure are conveniently calculated by using an equation of state applicable to both the phases.The major advantage of this approach is it's applicability over a wide range of pressures.High pressure vapor phase equilibria are often expressed in terms of K factors.The K factor for any component i is defined as

$$K_i = \frac{Y_i}{x_i} \qquad (4.1)$$

where y_i is the mole fraction of component i in the vapor phase and x_i is it's mole fraction in liquid phase.Equation (4.1) is rewrit in terms of fugacity coefficients.The condition for phase equilibrium is

$$f_i^v = f_i^l$$

where f_i is fugacity of i^{th} component in vapor and liquid phases respectively.Since for each phase fugacity coefficient is defined by

$$\varphi_i^v = \frac{f_i^v}{y_i p}$$

and $$\varphi_i^l = \frac{f_i^l}{x_i p}$$

From equation 4.1 it follows that

$$K_i = \frac{y_i}{x_i} = \frac{\varphi_i^l}{\varphi_i^v} \qquad 4.2$$

The φ_i^l and φ_i^v are to be calculated from some equation of state,which in our case is the Peng -Robinson equation of state. basic thermodynamic relation to express fugacity coefficient is given by

$$\ln \varphi_i = \frac{1}{RT} \int_v^\infty \left(\left(\frac{\partial p}{\partial n_i}\right)_{T,V,n_j} - \frac{RT}{V} \right) dv \; - \ln Z \qquad 4.3$$

where the compressibility factor Z is given by

$$Z = \frac{PV}{RT}$$

Mixing Rules :

The equations of state are generally expressed for pure components

first and then extended to mixtures.The expressions require the called mixing rules.These have the purpose of representing a property of a mixture in terms of the composition and properties of the pure components.

Typically a mixing rule expresses a mixture parameter Q_m in terms of compositions and pure components' parameters according to the relation

$$Q_m = \sum_i \sum_j y_i y_j Q_{ij} \qquad 4.4$$

In this equation the mole fraction $y_i y_j$ may apply to a liquid or vapor phase .Q_{ii} ,Q_{jj} involves properties of pure components i and j. The difficulty arises when interaction terms Q_{ij} are considered.Rules used to find Q_{ij} are called combining rules.
Van der Wall initially proposed a mixing rule of the type

$$Q_{ij} = \frac{Q_{ii}+Q_{jj}}{2}$$

$$Q_{ij} = (Q_{ii}Q_{jj})^{1/2}$$

These are very simple mixing rules.Better mixing rules are available now.The one which has been used in the present work is Wong and Sandler' mixing rule[11].

Outline of Wong and Sandler's mixinf rule with P-R equation of state:

The Peng -Robinson equation of state is given by

$$P = \frac{RT}{V-b} - \frac{a}{V^2 + 2bV - b^2}$$

where

$$b = \frac{0.07780\ RT_c}{P_c}$$

and

$$a = \frac{.45724\ R^2 T_c^2}{P_c}\ [1 + f\omega(1 - T_r^{1/2})]^2$$

where $f\omega = .37464 + 1.54226\omega - .26992\omega^2$

Based on the statistical thermodynamics the following mixing rule is given by Wong and Sandler [11].

The expressions for b_m and a_m are

$$b_m = \frac{Q}{(1-D)}$$

$$a_m = \frac{D}{(1-D)} Q$$

with Q and D defined as

$$Q = \sum_i \sum_j x_i x_j\ (b - a/RT)_{ij}$$

and

$$D = \sum_i x_i \frac{a_i}{b_i RT} + \frac{A_\infty^E}{CRT}$$

The fugacity coefficient is calculated from equation 4.3

For the P-R equation of state with any arbitrary set of mixing rules for a_m and b_m we have(*reference can be made Wong and*

Sandler[11]).

$$\ln \varphi_i = -\ln\left[\frac{P(\underline{V} - b_m)}{RT}\right] + \frac{1}{b_m}\left(\frac{\partial n b_m}{\partial n_i}\right)\left(\frac{P\underline{V}}{RT} - 1\right) + \frac{1}{2\sqrt{2}}\left(\frac{a_m}{b_m RT}\right)\left[\frac{1}{a_m}\left(\frac{\partial n^2 a_m}{\partial n_i}\frac{1}{n}\right) - \frac{1}{b_m}\left(\frac{\partial n b_m}{\partial n_i}\right)\right] \ln\left[\frac{\underline{V} + b_m(1-\sqrt{2})}{\underline{V} + b_m(1+\sqrt{2})}\right]$$

The partial derivatives of a_m and b_m are :

$$\frac{\partial n b_m}{\partial n_i} = \frac{1}{(1-D)}\left(\frac{\partial n^2 Q}{\partial n_i}\frac{1}{n}\right) - \frac{Q}{(1-D)^2}\left(1 - \frac{\partial n D}{\partial n_i}\right)$$

and

$$\frac{1}{RT}\left(-\frac{1}{n}\frac{\partial n^2 a_m}{\partial n_i}\right) = D\frac{\partial n b_m}{\partial n_i} + b_m\frac{\partial n D}{\partial n_i}$$

with the partial derivatives od Q and D given by

$$\left(\frac{1}{n}\frac{\partial n^2 Q}{\partial n_i}\right) = 2\sum x_j\left(b - \frac{a}{RT}\right)_{ij}$$

$$\frac{\partial n D}{\partial n_i} = \frac{a}{b_i RT} + \frac{\ln \gamma_{\infty i}}{C}$$

ith

$$\ln \gamma_{\infty i} = \frac{1}{RT}\frac{\partial n_i \underline{A}^E_\infty}{\partial n_i}$$

hough any free energy model can be used , here UNIQUAC has been sed for $\underline{A}^E_\infty / RT$.

$$\frac{\underline{A}^E}{RT} = \sum_i^N x_i \ln \frac{\varphi_i}{x_i} + \frac{z}{2} \sum_i^N q_i x_i \ln \frac{\vartheta_i}{\varphi_i} - \sum_i^N q_i x_i \ln\left(\sum_j^N \vartheta_j \tau_{ji}\right)$$

In this case ,the partial derivatives of $\underline{A}^E_\infty /RT$ with respect to the mole number of each species ,which is the logarithm of the specific activity coefficient , is given by

$$\ln\gamma_i = \ln \frac{\varphi_i}{x_i} + \frac{z}{2} q_i \ln\frac{\vartheta_i}{\varphi_i} + l_i - \frac{\varphi_i}{x_i} \sum_j^N x_j l_j - q_i \ln\left(\sum_j^N \vartheta_j \tau_{ji}\right) + q_i - q_i \sum_j^N \frac{\vartheta_j \tau_{ij}}{\sum_k^N \vartheta_k \tau_{kj}}$$

where

$$\varphi_i = \frac{r_i x_i}{\sum_k^N r_k x_k} \qquad \text{and} \qquad \vartheta_i = \frac{q_i x_i}{\sum_k^N q_k x_k}$$

here r and q are pure component parameters and the coordinatio number z=10.

$$\ln \tau_{ij} = -\frac{\nabla u_{ij}}{RT} \qquad \text{and} \qquad l_i = \frac{z}{2}(r_i - q_i) - (r_i - 1)$$

CHAPTER : 5
RESULTS AND DISCUSSION

The modified Gaussian elimination for single column was implemented on HP 9000 Supermini computer. The code was written in FORTRAN 77. The single column program has been tested with distillation problems.The details are presented below.

TEST OF THE PROGARM :

PROBLEM 1 (Naphtali and Sandholm 1971) :

This problem has been taken from Naphtali and Sandholm paper [5].The components to be separated were n-hexane and ethanol.The data for enthalpy calculation,critical properties were taken from Prausnitz and Reid [6]

The UNIQUACparameters were taken from Prausnitz et.al.[6] .The feed was vapor at $200^{o}F$,consisting of 50 moles of each component.The column contained ten plates with the feed entering the sixth.The reflux ratio was specified as 5.0 and the column was to recover 70% of the feed in the bottom product.This problem converged in seven iterations toan error criterion of the order of 10^{-5}.The results obtained were almost exactly matching with that of Naphtali and Sandholm's.

The nature of output file is shown on the next page.

PROBLEM-1

simulation of a distillation column

number of stages = 10

number of components = 2

number of reboilers = 1

type of reboiler : partial

total bottoms flow rate = 70.00000

number of condensers = 1

type of condenser : partial

reflux ratio = 5.000

results of simulation

error criterion = .500000E-04

vapour phase component flow rates

stage nos.	component no. 1	component no. 2	3	4	5	6
1	20.733	9.2674				
2	121.19	58.814				
3	116.89	61.975				
4	111.83	65.671				
5	105.82	70.048				
6	98.636	75.285				
7	36.516	27.990				
8	36.266	28.173				
9	35.482	28.747				
10	33.065	30.513				

liquid phase component flow rates

stage nos.	component no. 1	component no. 2	3	4	5	6
1	100.45	49.547				
2	96.157	52.708				
3	91.093	56.404				
4	85.091	60.780				
5	77.903	66.018				
6	65.784	68.722				
7	65.533	68.906				
8	64.749	69.479				
9	62.332	71.245				
10	29.267	40.733				

final vapour,liquid rate and temperature values

1	30.0000		150.0013	341.9
2	180.001		148.8648	341.9
3	178.865		147.4964	342.0
4	177.496		145.8714	342.1
5	175.871		143.9811	342.3
6	173.921		134.5059	342.6
7	64.5061		134.4387	342.6
8	64.4389		134.2385	342.6
9	64.2287		133.5775	342.7
10	63.5777		70.00000	343.0

stage	Temperature	l	v	x	y
1	341.85	150.00	30.00	.670E+00	.691E+00
2	341.93	148.86	180.00	.646E+00	.673E+00
3	342.00	147.50	178.86	.618E+00	.654E+00
4	342.12	145.87	177.50	.583E+00	.630E+00
5	342.30	143.92	175.87	.541E+00	.602E+00
6	342.57	134.51	173.92	.489E+00	.567E+00
7	342.58	134.44	64.51	.487E+00	.566E+00
8	342.61	134.23	64.44	.482E+00	.563E+00
9	342.71	133.58	64.23	.467E+00	.552E+00
10	343.05	70.00	63.58	.418E+00	.520E+00

condenser heat duty . = .476643E+07

reboiler heat duty = .212139E+07

number of iterations = 8

Problem 2 : In this problem, a distillation column with a large number of stages has been simulated. A 72 stage column separates a binary mixture into nearly pure products [3].
The feed is a saturated liquid at the feed pressure of 9.4 atm. and temperature of 363K.

The K values are calculated from the Wagner equation. The vapour enthalpies are calculated in terms of C_p , taken as a polynomial function of temperature. The latent heats are subtracted from the vapour enthalpies for computation of liquid enthalpies. The details of the output file are presented in the next page.

The problem converged in 11 iterations using the initialisation procedure with feed conditions as described in section 3.5. θ was taken to be 1.0 for all the steps in this case.

Problem :3

This problem deals with the simulation of a distillation column .A superheated vapour mixture consisting of five components is separated by a distillation column having an overhead partial condenser and a partial reboiler . The column consisting of thirteen stages and is operated at 400 psia.Feed enters the column at seventh stage at 105 ^{o}F .The feed consists of a mixture of C_1 to C_5 compounds .

The initial stage temperatures are set equal to the feed temperature . The total liquid and vapour flow rates are calculated by assuming equimolal flow .The

PROBLEM-2

simulation of a distillation column

number of stages = 72

number of components = 6

number of reboilers = 1

type of reboiler : partial

total bottoms flow rate = 121.00000

number of condensers = 1

type of condenser : partial

reflux ratio = 3.000

results of simulation

error criterion = .200846E-02

vapour phase component flow rates

stage nos.	component no. 1	2	3	4	5	6
1	114.60	.60033				
2	422.89	37.924				
3	222.01	162.18				
4	142.11	231.79				
5	129.16	247.56				
6	127.31	250.00				
7	127.05	250.35				
8	127.01	250.40				
9	127.01	250.41				
10	127.00	250.41				
11	127.00	250.41				
12	127.00	250.41				
13	127.00	250.41				
14	127.00	250.41				
15	127.00	250.41				
16	127.00	250.41				
17	127.00	250.41				
18	127.00	250.41				
19	127.00	250.41				
20	127.00	250.41				
21	127.00	250.41				
22	127.00	250.41				
23	127.00	250.41				
24	127.00	250.41				
25	127.00	250.41				
26	127.00	250.41				
27	127.00	250.41				
28	127.00	250.41				
29	127.00	250.41				
30	127.00	250.41				
31	127.00	250.41				
32	127.00	250.41				
33	127.00	250.41				

34	127.00	250.41
35	127.00	250.41
36	127.00	250.41
37	127.00	250.41
38	127.00	250.41
39	127.00	250.41
40	127.00	250.41
41	127.00	250.41
42	11.402	108.62
43	2.7032	121.52
44	.59288	124.89
45	.12749	125.64
46	.27254E-01	125.80
47	.61262E-02	125.84
48	.25100E-02	125.84
49	.19093E-02	125.85
50	.17962E-02	125.85
51	.17609E-02	125.85
52	.17347E-02	125.85
53	.16785E-02	125.85
54	.16857E-02	125.85
55	.16712E-02	125.85
56	.16796E-02	125.85
57	.16813E-02	125.85
58	.16801E-02	125.85
59	.16597E-02	125.85
60	.16517E-02	125.85
61	.16572E-02	125.85
62	.16554E-02	125.85
63	.16568E-02	125.85
64	.16601E-02	125.85
65	.16580E-02	125.85
66	.16588E-02	125.85
67	.16584E-02	125.85
68	.16559E-02	125.85
69	.16521E-02	125.85
70	.16509E-02	125.85
71	.16504E-02	125.85
72	.16606E-02	125.85

liquid phase component flow rates

stage nos.	component no.					
	1	2	3	4	5	6
1	308.29	37.324				
2	107.41	161.58				
3	27.514	231.19				
4	14.559	246.96				
5	12.707	249.40				
6	12.447	249.75				
7	12.410	249.80				
8	12.405	249.81				
9	12.405	249.81				
10	12.405	249.81				
11	12.405	249.81				
12	12.405	249.81				
13	12.405	249.81				
14	12.405	249.81				
15	12.405	249.81				
16	12.405	249.81				
17	12.405	249.81				
18	12.405	249.81				

19	12.405	249.81
20	12.405	249.81
21	12.405	249.81
22	12.405	249.81
23	12.405	249.81
24	12.405	249.81
25	12.405	249.81
26	12.405	249.81
27	12.405	249.81
28	12.405	249.81
29	12.405	249.81
30	12.405	249.81
31	12.405	249.81
32	12.405	249.81
33	12.405	249.81
34	12.405	249.81
35	12.405	249.81
36	12.405	249.81
37	12.405	249.81
38	12.405	249.81
39	12.405	249.81
40	12.405	249.81
41	11.402	229.61
42	2.7032	242.51
43	.59298	245.88
44	.12749	246.63
45	.27296E-01	246.80
46	.71791E-02	246.83
47	.39488E-02	246.84
48	.33730E-02	246.84
49	.32576E-02	246.84
50	.32321E-02	246.84
51	.31802E-02	246.84
52	.31585E-02	246.84
53	.31823E-02	246.84
54	.31805E-02	246.84
55	.31790E-02	246.84
56	.31801E-02	246.84
57	.31810E-02	246.84
58	.31748E-02	246.84
59	.31726E-02	246.84
60	.31724E-02	246.84
61	.31729E-02	246.84
62	.31735E-02	246.84
63	.31744E-02	246.84
64	.31748E-02	246.84
65	.31749E-02	246.84
66	.31748E-02	246.84
67	.31740E-02	246.84
68	.31732E-02	246.84
69	.31727E-02	246.84
70	.31724E-02	246.84
71	.31745E-02	246.84
72	.15975E-02	121.00

final vapour,liquid rate and temperature values

1	115.200	345.6153	266.7
2	460.815	268.9984	288.0
3	384.198	259.7038	317.8
4	373.904	261.5161	326.7
5	376.716	262.1107	328.1
6	377.311	262.1977	328.3
7	377.398	262.2098	328.3

stage	Temperature	l	v	x	y
8		377.410		262.2115	328.3
9		377.412		262.2117	328.3
10		377.412		262.2118	328.3
11		377.412		262.2118	328.3
12		377.412		262.2118	328.3
13		377.412		262.2118	328.3
14		377.412		262.2118	328.3
15		377.412		262.2118	328.3
16		377.412		262.2118	328.3
17		377.412		262.2118	328.3
18		377.412		262.2118	328.3
19		377.412		262.2118	328.3
20		377.412		262.2118	328.3
21		377.412		262.2118	328.3
22		377.412		262.2118	328.3
23		377.412		262.2118	328.3
24		377.412		262.2118	328.3
25		377.412		262.2118	328.3
26		377.412		262.2118	328.3
27		377.412		262.2118	328.3
28		377.412		262.2118	328.3
29		377.412		262.2118	328.3
30		377.412		262.2118	328.3
31		377.412		262.2118	328.3
32		377.412		262.2118	328.3
33		377.412		262.2118	328.3
34		377.412		262.2118	328.3
35		377.412		262.2118	328.3
36		377.412		262.2118	328.3
37		377.412		262.2118	328.3
38		377.412		262.2118	328.3
39		377.412		262.2118	328.3
40		377.412		262.2118	328.3
41		377.412		241.0110	328.3
42		120.018		245.2142	336.6
43		124.221		246.4738	338.7
44		125.480		246.7620	339.2
45		125.768		246.8240	339.3
46		125.830		246.8385	339.3
47		125.843		246.8426	339.3
48		125.847		246.8437	339.3
49		125.848		246.8438	339.3
50		125.848		246.8437	339.3
51		125.848		246.8434	339.3
52		125.847		246.8435	339.3
53		125.847		246.8427	339.3
54		125.847		246.8427	339.3
55		125.847		246.8434	339.3
56		125.847		246.8433	339.3
57		125.847		246.8439	339.3
58		125.848		246.8434	339.3
59		125.847		246.8435	339.3
60		125.847		246.8441	339.3
61		125.848		246.8443	339.3
62		125.848		246.8441	339.3
63		125.848		246.8447	339.3
64		125.849		246.8456	339.3
65		125.850		246.8456	339.3
66		125.850		246.8457	339.3
67		125.850		246.8452	339.3
68		125.849		246.8456	339.3
69		125.849		246.8465	339.3
70		125.850		246.8452	339.3
71		125.849		246.8451	339.3
72		125.849		121.0016	339.3

1	266.66	345.62	115.20	.892E+00	.995E+00
2	299.03	269.00	460.82	.399E+00	.918E+00
3	317.80	258.70	384.20	.106E+00	.578E+00
4	326.69	261.52	373.90	.557E-01	.380E+00
5	329.10	262.11	376.78	.485E-01	.343E+00
6	328.30	262.20	377.31	.475E-01	.337E+00
7	328.33	262.21	377.40	.473E-01	.337E+00
8	328.33	262.21	377.41	.473E-01	.337E+00
9	328.33	262.21	377.41	.473E-01	.337E+00
10	328.33	262.21	377.41	.473E-01	.337E+00
11	328.33	262.21	377.41	.473E-01	.337E+00
12	328.33	262.21	377.41	.473E-01	.337E+00
13	328.33	262.21	377.41	.473E-01	.337E+00
14	328.33	262.21	377.41	.473E-01	.337E+00
15	328.33	262.21	377.41	.473E-01	.337E+00
16	328.33	262.21	377.41	.473E-01	.337E+00
17	328.33	262.21	377.41	.473E-01	.337E+00
18	328.33	262.21	377.41	.473E-01	.337E+00
19	328.33	262.21	377.41	.473E-01	.337E+00
20	328.33	262.21	377.41	.473E-01	.337E+00
21	328.33	262.21	377.41	.473E-01	.337E+00
22	328.33	262.21	377.41	.473E-01	.337E+00
23	328.33	262.21	377.41	.473E-01	.337E+00
24	328.33	262.21	377.41	.473E-01	.337E+00
25	328.33	262.21	377.41	.473E-01	.337E+00
26	328.33	262.21	377.41	.473E-01	.337E+00
27	328.33	262.21	377.41	.473E-01	.337E+00
28	328.33	262.21	377.41	.473E-01	.337E+00
29	328.33	262.21	377.41	.473E-01	.337E+00
30	328.33	262.21	377.41	.473E-01	.337E+00
31	328.33	262.21	377.41	.473E-01	.337E+00
32	328.33	262.21	377.41	.473E-01	.337E+00
33	328.33	262.21	377.41	.473E-01	.337E+00
34	328.33	262.21	377.41	.473E-01	.337E+00
35	328.33	262.21	377.41	.473E-01	.337E+00
36	328.33	262.21	377.41	.473E-01	.337E+00
37	328.33	262.21	377.41	.473E-01	.337E+00
38	328.33	262.21	377.41	.473E-01	.337E+00
39	328.33	262.21	377.41	.473E-01	.337E+00
40	328.33	262.21	377.41	.473E-01	.337E+00
41	328.33	241.01	377.41	.473E-01	.337E+00
42	336.58	245.21	120.02	.110E-01	.950E-01
43	338.72	246.47	124.22	.241E-02	.218E-01
44	339.20	246.76	125.48	.517E-03	.472E-02
45	339.30	246.82	125.77	.111E-03	.101E-02
46	339.33	246.84	125.83	.291E-04	.217E-03
47	339.33	246.84	125.84	.160E-04	.487E-04
48	339.33	246.84	125.85	.137E-04	.199E-04
49	339.33	246.84	125.85	.132E-04	.152E-04
50	339.33	246.84	125.85	.131E-04	.143E-04
51	339.33	246.84	125.85	.129E-04	.140E-04
52	339.33	246.84	125.85	.128E-04	.122E-04
53	339.33	246.84	125.85	.129E-04	.133E-04
54	339.33	246.84	125.85	.129E-04	.134E-04
55	339.33	246.84	125.85	.129E-04	.133E-04
56	339.33	246.84	125.85	.129E-04	.133E-04
57	339.33	246.84	125.85	.129E-04	.134E-04
58	339.33	246.84	125.85	.129E-04	.134E-04
59	339.33	246.84	125.85	.129E-04	.132E-04
60	339.33	246.84	125.85	.129E-04	.131E-04
61	339.33	246.84	125.85	.129E-04	.131E-04
62	339.33	246.84	125.85	.129E-04	.132E-04
63	339.33	246.84	125.85	.129E-04	.132E-04
64	339.33	246.85	125.85	.129E-04	.132E-04
65	339.33	246.85	125.85	.129E-04	.132E-04
66	339.33	246.85	125.85	.129E-04	.132E-04

67	339.33	246.85	125.85	.129E-04	.132E-04
68	339.33	246.85	125.85	.129E-04	.132E-04
69	339.33	246.85	125.85	.129E-04	.131E-04
70	339.33	246.85	125.85	.129E-04	.131E-04
71	339.33	246.85	125.85	.129E-04	.131E-04
72	339.33	121.00	125.85	.132E-04	.132E-04

condenser heat duty = .808654E+07

reboiler heat duty = .369743E+07

number of iterations = 11

compositions of each liquid and vapour stream are set equal to the composition of the feed .

PROBLEM-3

simulation of a distillation column

number of stages = 13

number of components = 5

number of reboilers = 1

type of reboiler : partial

total bottoms flow rate = 280.00000

number of condensers = 1

type of condenser : partial

reflux ratio = 1.890

results of simulation

error criterion = .915200E-02

vapour phase component flow rates

stage nos.	component no. 1	2	3	4	5	6
1	160.00	358.92	1.0815	.76531E-02	.19824E-02	
2	225.12	1265.3	12.379	.29043E-01	.66728E-02	
3	189.12	1278.5	35.376	.34913E-01	.70768E-02	
4	182.95	1200.4	87.018	.67525E-01	.73121E-02	
5	180.60	1057.6	179.77	.36424	.90651E-02	
6	178.74	887.99	297.82	2.0406	.47394E-01	
7	177.25	742.95	397.54	8.7394	.57027	
8	15.674	265.37	190.73	4.0287	.26479	
9	3.7214	236.18	246.20	4.9601	.32603	
10	.83093	186.57	302.47	5.9003	.38451	
11	.20324	133.75	358.65	7.1504	.44980	
12	.85631E-01	86.285	407.83	9.7464	.58438	
13	.55937E-01	47.974	437.17	17.686	1.4334	

liquid phase component flow rates

stage nos.	component no. 1	2	3	4	5	6
1	65.121	906.38	11.297	.21720E-01	.47174E-02	
2	29.124	919.58	34.294	.27958E-01	.51408E-02	
3	22.953	841.44	85.936	.61765E-01	.53772E-02	
4	20.600	698.65	178.69	.36388	.71595E-02	
5	18.739	529.07	296.80	2.0406	.47060E-01	
6	17.255	384.03	396.46	8.7394	.57027	
7	15.674	276.45	429.64	29.029	5.2648	
8	3.7218	247.26	485.12	29.960	5.3260	
9	.83456	197.65	541.39	30.900	5.3845	
10	.22574	144.83	597.57	32.150	5.4498	

11	.11383	97.366	646.75	34.746	5.5844
12	.83988E-01	59.055	676.09	42.686	6.4334
13	.28721E-01	11.081	238.92	25.000	5.0000

final vapour,liquid rate and temperature values

1	520.010	982.8267	11.66
2	1502.84	983.0302	28.40
3	1503.04	950.3916	34.77
4	1470.40	898.3143	43.80
5	1418.32	846.6976	57.77
6	1366.70	807.0547	73.65
7	1327.05	756.0645	88.23
8	476.064	771.3887	105.7
9	491.389	776.1581	117.7
10	496.155	780.2266	128.7
11	500.205	784.5571	138.6
12	504.529	784.3445	147.5
13	504.317	280.0283	157.4

stage	Temperature	l	v	x	y
1	11.66	982.83	520.01	.663E-01	.308E+00
2	28.40	983.03	1502.84	.296E-01	.150E+00
3	34.77	950.39	1503.04	.242E-01	.126E+00
4	43.80	898.31	1470.40	.229E-01	.124E+00
5	57.77	846.70	1418.32	.221E-01	.127E+00
6	73.65	807.05	1366.70	.214E-01	.131E+00
7	88.23	756.06	1327.05	.207E-01	.134E+00
8	105.72	771.39	476.06	.482E-02	.329E-01
9	117.75	776.16	491.39	.108E-02	.757E-02
10	128.66	780.23	496.15	.389E-03	.167E-02
11	138.62	784.56	500.20	.145E-03	.406E-03
12	147.48	784.34	504.53	.107E-03	.170E-03
13	157.36	280.03	504.32	.103E-03	.111E-03

condenser heat duty = .474591E+07

reboiler heat duty = .292744E+07

number of iterations = 7

CONCLUSIONS

The Gauss elimination has been suitably modified to solve the Jacobian in case of single columns. The inner sparsity of the blocks of the block-tridiagonal Jacobian has been exploited. The equation of state has been sucessfully incorporated to take care of highly non-ideal systems.The beauty of this approach tc calculate equilibrium constants [K] is this that it does not requ experimental data.It needs only critical parameters.

Computer codes have been written to implement the modified Gauss elimination technique and equation of state approach. The cod worked well for single distillation columns.

NOMENCLATURE

A, B, C	(2C+1)x(2C+1) matrices
a_{il}	nonzero element at location (i,l) of **M**
b	vector of order m
C	number of components
C_p	specific heat
d	scalar
d	array of order Nx(2C+1)
E_j	enthalpy balance discrepancy function for the j-th stage
$e^{(n+1)}$	(n+1) order unit vector, $[\ 0,0,....0,1]^T$
F	vector of discrepancy functions of order N(2C+1)
F_j	vector of discrepancy functions of order (2C+1)
$f_{j,i}$	flow rate of component i in the feed to j-th stage
g	array of order (4C+2)x(4C+3)
$H_{j,i}$	enthalpy of vapour of component i leaving the j-th stage
h	scalar
$h_{j,i}$	enthalpy of liquid of component i leaving the j-th stage
h	array to store upper-triangular matrix, order Nx(2C+1)x(3C+2)
h_L	array of dimension (2C+1)x(2C+1) to store lower off-tridiagonal blocks
h_U	array of dimension (2C+1)x(2C+1) to store upper off-tridiagonal blocks
I_c	identity matrix of order CxC
J	Jacobian of order N(2C+1)xN(2C+1)

$K_{j,i}$ equilibrium constant of component i at the j-th stage

$l_{j,i}$ component liquid flow rate for component i leaving j-th stage

L_j total flow rate of liquid leaving the j-th stage

$M_{j,i}$ material balance discrepancy function for component i leaving stage j

M square matrix of order mxm

m, m' matrices of order NxN

N number of stages

n number of equations

0_C null column vector of order C

0_{CC} null matrix of order CxC

P vector of order n

$Q_{j,i}$ discrepancy functions for phase equilibrium relations for component i leaving stage j

q_j heat supplied/withdrawn from the j-th stage

$R_{a,b}$ fraction of liquid leaving stage b for stage a

$r_{L_{j-1}}$ fraction of liquid stream from (j-1)-th stage reaching j-th stage

$r_{V_{j+1}}$ fraction of vapour stream from (j+1)-th stage reaching j-th stage

S_j fraction of the vapour side-stream from j-th stage

s_j fraction of the liquid side-stream from j-th stage

T_j temperature of the j-th stage

V_j total flow rate of vapour leaving the j-th stage

$v_{j,i}$ component vapour flow rate for component i leaving j-th stage

X	a nonzero element
$\mathbf{X}$	vector of variables of order Nx(2C+1)
$\mathbf{X}_c$	column vector of nonzero elements of order C
$\mathbf{X}_{cc}$	matrix of nonzero elements of order CxC
$\mathbf{X}_j$	vector of variables of order (2C+1)
$\mathbf{X}^*$	desired solution of equation system
$\mathbf{X}^o$	initial guess vector
$x_{j,i}$	mole fraction of component i in the vapour leaving the j-th stage
$y_{j,i}$	mole fraction of component i in the vapour leaving the j-th stage
ϑ	convergence criterion
$\eta_{j,i}$	Murphree efficiency of component i at j-th stage
θ	fraction of the Newton-Raphson correction
a	equation of state energy parameter
$\underline{A}$	molar helmholtz free energy
b	equation of state excluded papameter
B	second virial coefficient
F	arbitrary function
$\underline{V}$	molar volume
ϕ	fugacity coefficient
γ	activity coefficient
r,q,τ	uniquac papameters

REFERENCES

[1] Broyden C.G., A class of methods for solving nonline simultaneous equations. *Math. Comp.*,19,577 (1965).

[2] Dutt I., Studies on the simulation of separation proces *M. Tech Thesis,* I.I.T., Kanpur (1990).

[3] Henley E.J. and J.D.Seader, *Equilibrium Stage Separ Operations in Chemical Engineering*. John Wiley and sons Inc. New York (1981).

[4] Kubicek M., Algorithm 470,Linear systems with alm Tridiagonal matrix., *Communs. ACM*,16(12),760 (1973).

[5] Naphtali L.M. and D.P.Sandholm, Multicomponent separatio calculations by linearization. *AIChE. J.* 17(1),148 (1971).

[6] Reid J. C., J. M. Prausnitz, B. Poling, *Properties of Ga and Liquids.* Mc GrawHill Book Co., New York (1988).

[7] Venakata Ramana R., An efficient method of solution for problems of interlinked separation processes, *M. Tech Thesis,* I. I. T., Kanpur (1991).

[8]Wayburn T.L. and J.D.Seader,Homotopy methods for computer aided process design . *Comput chem Engng* 11(1) , 7 (1981).

[9] Jelinek j., V. Hlavacek and M. Kubicek , Calculation of multistage counter current separation processes - I. Multicomponent multistage rectification by differentiation with respect to an actual parameter . *chem. Engmg Sci. 28 1555 (1973).*

[10] Wayburn T.L. and J.D.Seader, Solutions of systems interlinked distillation columns by differential homotopy continuation methods. - *Foundations of Computer Aide Chemical Process Design.* (Westerberg A.W. and .H.Chien,eds CACHE (1984).

[11] Wong D.S.H. and Sandler.S.I. A theoretically correct mi rule for cubic equation of state. *AICHE VOL 38 NO 5 PAGE NO 671 1992.*

[12] Charles D. Holland , Fundamentals of Multicomponent Distillation .

[13] Philip T. Eubank,Gour Shiarn Shyu,and Nishawn S.M.hanif New procedures for application of the Wong -Sandler Mixing rules to the prediction of Vapor Liqid equilibri *.Eng.Chem.Res 1995 ,34 314-323 .*

APPENDIX

The elements of the A, B and C blocks :

$$A_j = \left| \frac{\partial F_{j,i}}{\partial X_{j-1,k}} \right|$$

$$\frac{\partial M_{j,i}}{\partial l_{j-1,k}} = r_{L_{j-1}} \delta_{i,k} \quad ;$$

$$\delta_{i,k} = 1 \quad , \text{ if } i = k$$
$$\delta_{i,k} = 0 \quad , \text{ if } i \neq k$$

$$\frac{\partial M_{j,i}}{\partial v_{j-1,k}} = 0 \quad ; \quad \frac{\partial M_{j,i}}{\partial T_{j-1}} = 0$$

$$\frac{\partial Q_{j,i}}{\partial l_{j-1,i}} = 0 \quad ; \quad \frac{\partial Q_{j,i}}{\partial v_{j-1,i}} = 0 \quad ; \quad \frac{\partial Q_{j,i}}{\partial T_{j-1}}$$

$$\frac{\partial E_j}{\partial l_{j-1,i}} = r_{L_{j-1}} h_{j-1,i} \quad ; \quad \frac{\partial E_j}{\partial v_{j-1,i}} = 0$$

$$\frac{\partial E_j}{\partial T_{j-1}} = r_{L_{j-1}} \sum l_{j-1,i} \frac{\partial h_{j-1,i}}{\partial T_{j-1}}$$

$$B_j = \left| \frac{\partial F_{j,i}}{\partial X_{j,k}} \right|$$

$$\frac{\partial M_{j,i}}{\partial l_{j,k}} = -(1+s_j)\delta_{j,k}$$

$$\frac{\partial M_{j,i}}{\partial v_{j,k}} = -(1+s_j)\delta_{j,k} \quad ; \quad \frac{\partial M_j}{\partial T_j} = 0$$

$$\frac{\partial Q_{j,i}}{\partial l_{j,k}} = \frac{\eta_{j,k} K_{j,i}}{L_j^2} (L_j \delta_{i,k} - l_{j,i}) + \frac{\eta_{j,i} l_{j,i}}{L_j} \frac{\partial K_{j,i}}{\partial l_{j,k}} \delta_{i,k}$$

$$\frac{\partial Q_{j,i}}{\partial v} = \frac{\eta_{j,i} l_{j,i}}{L} \frac{\partial K_{j,i}}{\partial v_{j,k}} - \frac{1}{V_j^2} (V_j \delta_{i,k} - v_{j,i})$$

$$\frac{\partial Q_{j,i}}{\partial T_j} = \frac{\eta_{j,i} l_{j,i}}{L_j} \frac{\partial K_{j,i}}{\partial T_j}$$

$$\frac{\partial E_j}{\partial l_{j,i}} = -(1+s_j) h_{j,i} \quad ; \quad \frac{\partial E_j}{\partial v_{j,i}} = -(1+s_j) H_{j,i}$$

$$\frac{\partial E_j}{\partial T_j} = -(1+s_j) \sum_{i=1}^{C} l_{j,i} \frac{\partial h_{j,i}}{\partial T_j} - (1+s_j) \sum_{i=1}^{C} v_{j,i} \frac{\partial H_{j,i}}{\partial T_j}$$

$$C_j = \left| \frac{\partial F_{j,i}}{\partial X_{j+1,k}} \right|$$

$$\frac{\partial M_{j,i}}{\partial l_{j+1,k}} = R_{j,j+1} \delta_{j,k} = 0$$

$$\frac{\partial M_{j,i}}{\partial v_{j+1,k}} = r_{V_{j+1}} \delta_{j,k} \quad ; \quad \frac{\partial M_{j,i}}{\partial T_{j+1}} = 0$$

$$\frac{\partial Q_{j,i}}{\partial l_{j+1,k}} = 0$$

$$\frac{\partial Q_{j,i}}{\partial v_{j+1,k}} = \frac{(1-\eta_{j,i})\left\{ r_{V_{j+1}} V_{j+1} + \sum_{kk} R_{j,kk} V_{kk}) r_{V_{j+1}} \delta_{j,k} - (r_{V_{j+1}} v_{j+1} + \sum Rv)\ r_{V_{j+1}} \right\}}{\left\{ (r_{V_{j+1}} V_{j+1} + \sum_{kk} R_{j,kk} V_{kk})^2 \right\}}$$

$$\frac{\partial Q_{j,i}}{\partial T_{j+1}} = 0$$

$$\frac{\partial E_j}{\partial l_{j+1,k}} = R_{j,j+1} h_{j+1,k} = 0$$

$$\frac{\partial E_j}{\partial v_{j+1,k}} = r_{V_{j+1}} H_{j+1,k}$$

$$\frac{\partial E_j}{\partial T_{j+1}} = r_{V_{j+1}} \sum v_{j+1,i} \frac{\partial H_{j+1,i}}{\partial T_{j+1}}$$